ELSEVIER
CrossMark
Procedia Computer Science
Volume 29, 2014, Pages 1048–1056
ICCS 2014. 14th International Conference on Computational Science
Procedia Computer Science

Translation of graph-based knowledge representation in multi-agent system

Leszek Kotulski[1] , Adam Sędziwy[1] and Barbara Strug[1,2]

[1] Department of Applied Computer Science,
AGH University of Science and Technology
Al. Mickiewicza 30, 30 059 Krakow, Poland
[2] Department of Physics, Astronomy and Applied Computer Science,
Jagiellonian University, Reymonta 4, Krakow, Poland
{kotulski,sedziwy,bstrug}@agh.edu.pl

Abstract
Agents provide a feasible mean for maintaining and manipulating large scale data. This paper deals with the problem of information exchange between different agents. It uses graph based formalism for the representation of knowledge maintained by an agent and graph transformations as a mean of knowledge exchange. Such a rigorous formalism ensures the cohesion of graph-based knowledge held by agents after each modification and exchange action. The approach presented in this paper is illustrated by a case study dealing with the problem of personal data held in different places (maintained by different agents) and the process of transmitting such information [1].

Keywords:

1 Introduction

The size and type of data stored by different institutions and corporations is becoming increasingly complex as well as grows in size. In many cases it becomes infeasible to be stored and processed in one database. Moreover in many cases there are other factors preventing storage and processing of data in some databases (like legal and data protection requirements). On the other hand distributed model of computing is becoming more and more popular, especially with a rapid development of the Internet and availability of distributed development platforms. Moreover the information becomes also distributes. It is now very likely for different aspects of knowledge about some entity to be held in different places (or by different agents). As the information can be distributed many tasks can require cooperation of autonomous agents exchanging pieces of information they have or asking for a missing information from other agent.

[1]Financial support for this study was provided from resources of National Center for Research and Development, the grant number NCBiR 0021/R/ID2/2011/01.

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

doi:10.1016/j.procs.2014.05.094

With the knowledge maintained by different agents becomes more and more complex a representation capable of capturing the inherent complexity of knowledge is also required. Moreover there is also a need for such a representation to be able not only to maintain the elements of the knowledge but also more and more complex relations within the knowledge.

To represent complex objects in different domains of computer science graphs are very often used [15]. Their ability to represent the structure of data as well as the different types of relations between its elements makes them particularly useful as the mean of representing complex information. This trend can be also observed in the emergence of graph databases, which are often proposed as better suited for complex knowledge systems then traditional relational databases [14].

A new approach to graph based knowledge exchange/transmission, proposed in this paper, operates on the distributed, possibly heterogenous information, represented in graph form and uses graph transformations as a basis for knowledge transmission. It is based on earlier research in the domain of application of the theory of formal languages in different domains such as computer aided design [6, 7, 11, 17, 16] and distributed model of computation [5, 15, 9, 10, 12]. In particular graph grammars [1, 6, 8, 13] and grammar systems [3, 2, 4] were used as the inspiration for this research as well as graph cohesion and cooperation.

2 Graph-based knowledge representation and modification

In the proposed graph-based approach, all pieces of knowledge are represented by labelled graphs. The benefits of using graphs as the representation of knowledge include several facts. Firstly, graphs are simple mathematical objects which have graphical representations which allows for visualisation. There is also a number of efficient algorithms for processing graphs and graphs can be equipped with a logical semantics. Moreover they allow for uniform formal description of knowledge, its elements, relations between these elements as well as form a good basis for formal description of knowledge processing.

2.1 Knowledge representation

Let a given graph database contains information about a number of objects, each having some properties and being related to other objects. Let Σ_i be the set of node and edge labels, let A be the set of node and label attributes and D_a be a set of possible values of an attribute $a \in A$. Thus let G_i be a graph representing knowledge held by an agent A_i, The G_i is defined over Σ_i in the following way

Definition 1. *A labelled and attributed graph over $\Sigma = \Sigma_E \cup \Sigma_V$ is a system $G = (V, E, lab, att)$, where:*

- *V is a finite set of nodes; $E \subseteq V$ is a finite set of edges,*
- *$lab = (lab_V, lab_E)$ is a labeling function, where: $lab_V : V \to \Sigma_V$, $lab_E : E \to \Sigma_E$;*
- *$att = (att_V, att_E)$ is an attributing function, where: $att_V : \Sigma_V \to P(A_V)$, $att_E : \Sigma_E \to P(A_E)$ and A_V, A_E are sets of node and edge attributes respectively;*

The family of all labelled and attributed graphs is denoted by $\mathscr{H}$.

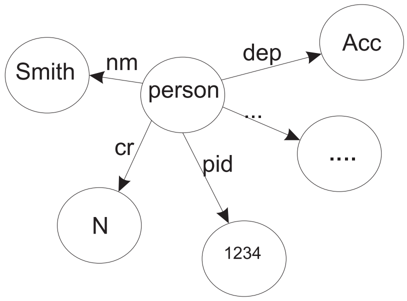

Figure 1: An example of a part of graph database

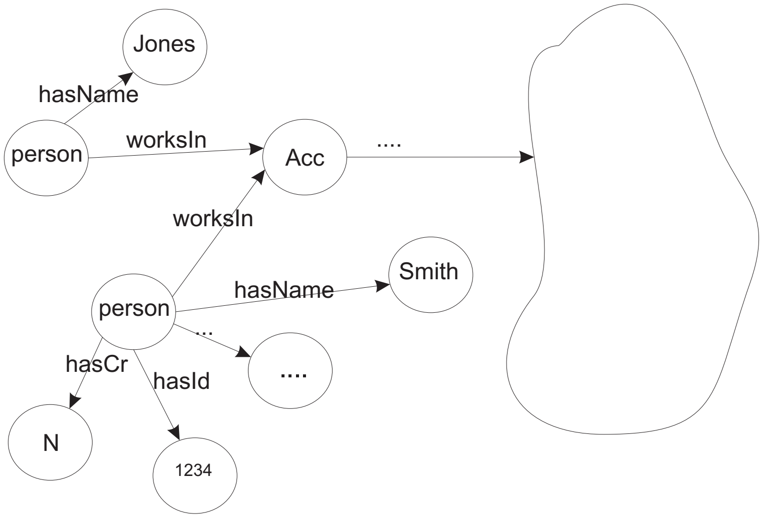

Figure 2: An example of a part of graph database using RDF graphs

Let $val = (val_V, val_E)$ be a mapping assigning a value, where: $val_V : A_V \times V \to D_a$, $val_E : A_E \times E \to D_a$ assign a value to a given attribute a of a node/edge, D_a is called a domain of a node/edge attribute. The set of all val functions will be denoted as VAL

Definition 2. *An instance of a labelled and attributed graph over* $\Sigma = \Sigma_E \cup \Sigma_V$ *is a pair* $\hat{G} = (G, val) \in \mathscr{H} \times VAL$*. The family of all instances* $\hat{G}$ *will be denoted as* $\hat{\mathscr{H}}$*.*

Such a general notion of graph representation of knowledge does not put any strict constraints on the type of graph used in a given graph database. For example RDF graphs, which become increasingly popular as a mean to represent knowledge online, can be used. In this case a set of nodes would consist of resources (objects) represented in a given graph, with Σ_V be the set of names of these objects, while Σ_E would consist of the names of RDF properties. In case of RDF graphs the set of attributes A is empty. Two examples of possible types of graphs representing personal information of employees in a company are presented in Fig. 2 and 1. An example with a graph database using a nonempty set of attributes is presented in Fig. 3a and an instance of this graph with all attributes having assigned values is depicted in Fig. 3b.

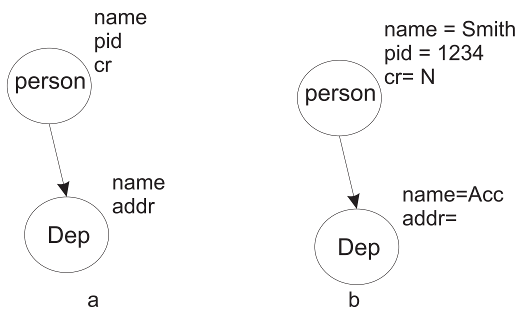

Figure 3: An example of a graph using nonempty set of attributes (a) and its instance (b)

2.2 Knowledge modification

Let A_i be an agent and G_i be the graph representing the knowledge kept by A_i. Let Q be a special set of query symbols.The knowledge can be modified by a graph transformation system.

Definition 3. *Let $\hat{\mathscr{H}}$ be a family of instances of labeled, attributed graphs, then a grammar production is a tuple of the form $p = (l, r, \pi)$, where:*

1. *l is a left side of the production, being a labeled graph over Σ*
2. *r is a labelled graph called a right side of the production and defined over $\Sigma \cup Q$*
3. *$\pi : \hat{\mathscr{H}} \to \{TRUE, FALSE\}$, is a predicate determining the production applicability.*

A production p will be also denoted as $p : l \to_\pi r$ or in the abbreviated form: $p : l \longrightarrow r$.

Application of the production $p : l \longrightarrow r$ to a given graph G (for example to the graph shown in Fig. 1) is accomplished in following steps: a subgraph $L \subseteq G$, isomorphic with the left side of a production, i.e., with l, is taken. Next L is removed from G and replaced by the right side of the production, i.e., by r. Finally one has to fix all edges connecting previously $G - L$ with L. Those edges have to be either removed or "reattached" to r.

If the graph r is defined over Σ only, i.e. it does not contain labels from the set of query symbols such a production is local and its application can be controlled locally. In case of graph r containing a node labelled by an element from Q, i.e. by a query symbol the cooperation process has to be triggered by a global control mechanism.

3 Knowledge transmission between agents through graph transformations

While graphs can represent a piece of knowledge graph transformations provide a formal and well defined model for describing the process of modifying and updating a data as described above. Yet, in modern times data is often to big to be kept in one place. Moreover in some cases keeping all data in one place may not even be desired or legal (for example the way personal data is kept is controlled by the data protection laws). Thus the knowledge is often distributed and different pieces (although not necessary disjoint) are kept by different agents. In this paper we want to extend the model of graph transformations to allow not only for describing the modifications of knowledge held by single agent but also to introduce a mechanism for formal description of knowledge transmission.

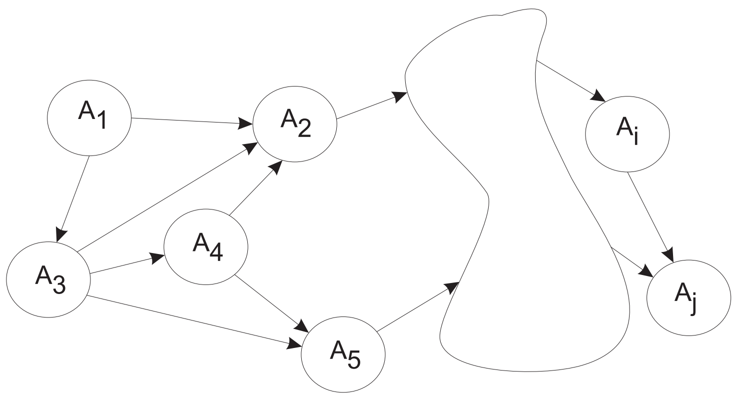

Figure 4: A set of agents and information channels between them

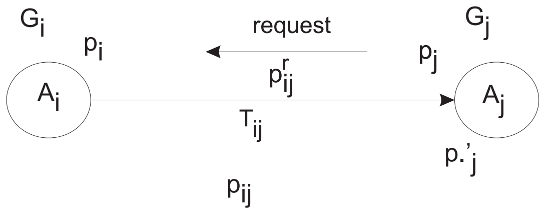

Figure 5: Communication between agents over a communication channel

Let there is a multi agent system consisting of a number of agents, denoted by A_i, and communication channels)depicted in Fig. 4. Let A_i, A_j be agents, where A_i has a piece of information required by A_j. Let G_i, G_j be graphs kept by these agents. Let Q be a set of query symbols allowing for the cooperation between agents.

Let T_{ij} be a channel through which a knowledge can be passed from agent A_i to agent A_j. Each channel is directed, and the direction (depicted as arrow in Fig. 4) shows the way knowledge can be sent. It also implies that the request for some piece of knowledge is sent in the opposite direction. If both agents can sent some piece of knowledge to each other two channels have to be defined.

Let GT_{IJ} be the graph transformation system associated with this channel responsible for translating the format of knowledge held by agent A_i into format acceptable by agent A_j. As the agent A_j can sent request for knowledge to be provided the graph transformation system GT^r_{IJ}, also associated with the channel T_{ij} is responsible for the translation of request.

Thus a production of such a transformation channel is defined in the similar way as for the agents but with the following differences:

- a graph l, being the left side of the production is of the same type as the graph representing the knowledge kept by agent A_i and is defined over the set $\Sigma_i \cup Q$,
- a graph r, being the right side of the production is of the same type as the graph representing the knowledge kept by agent A_j and is defined over the set $\Sigma_j \cup Q$.

For the GT^r_{IJ} transformation system in the above the indices i and j have to be exchanged.

Let G_i be defined over $|Sigma_i \cup Q$. Let p_j be a production $l \rightarrow r$, such that r contains at least one node labelled by a query symbol. Then the global control system triggers a production

p_{ij}^r from the graph transformation system associated with a channel T_{ij}. responsible for handling requests Such a production takes as its left side a graph being the right side of the production p_j and replaces it be a graph r defined over the set$\Sigma_i \cup Q$, that is a graph defined over the same set as the graph database kept by agent A_i. The application of this production triggers a production p_i associated with p_{ij}^r of the transformation system GT_i. This production takes the right side of the previous production as its left side, l_j. The application of this production consists of finding a subgraph of $G)i$ isomorphic with $l_j - x$,where x is a subgraph containing query symbols. Then the actual knowledge from G_i is used to replace the query symbol. The applicability predicate of this production is set to TRUE if such a match is found, and to FALSE otherwise. If the production is applicable it triggers an associated production p_{ij} from the transformation system associated with the transmission channel which is responsible for returning the requested information. And finally a production p_j' is applied, which has the left side identical with the production p_j and uses the right side of the production p_{ij} as its right side.

Thus, it can be said that the graph transformation system associated with a channel is responsible for translating the format of knowledge from the one used by the querying agent to the one used by the receiving agent, but it also has to preserve the query symbol.

4 Case study

Let us consider a situation in which two agents A and B are responsible for maintaining two graph databases. Agent A deals with a company database containing employees personal data, while B is responsible for some security/legal database containing, among others, knowledge about persons criminal record. The format of graphs used by A and B ate depicted in Fig. 6 and Fig. 7, respectively. As it can be noticed, the format in which person is identified in both systems is different. Moreover, let's assume thet there is a transmission channel T_{BA}, thus a knowledge from the database maintained by B can be transmitted to A.

The agent A applies a production p_1, depicted in Fig. 8, adding an information on a persons criminal record (cr) which is unavailable, thus the actual information is referred by a query symbol q. The presence of the query symbol indicates that this production cannot be applied locally but a cooperation must be established with the agent associated with this symbol (in our case B).In the next step a production from the transformation system GT_{BA}^r is triggered (this production is depicted in Fig. 9). It is responsible for converting the request into a format in which knowledge is held by agent B. The application of this production is followed by a production from the local transformation system of agent B, which results in replacing the query symbol by an actual information (in this case 'N', meaning a person requested has no criminal record). Finally productions depicted in Fig. 11 are responsible for actually passing the requested knowledge from B to A.

The mechanism described here ensures that each agent always received information in format it accepts. Moreover a given agent does not need to know anything about knowledge format used by other agents in a multi-agent system as the appropriate transformation is catered for by the channel connecting agents.

5 Conclusions and future research

This paper presents a new approach to the transmission of knowledge between agents. It uses a graph bases representation for agents, which is becoming very popular, especially with the

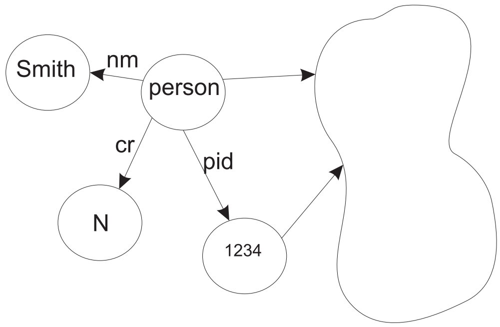

Figure 6: Small fragment of a graph database maintained by an agent A

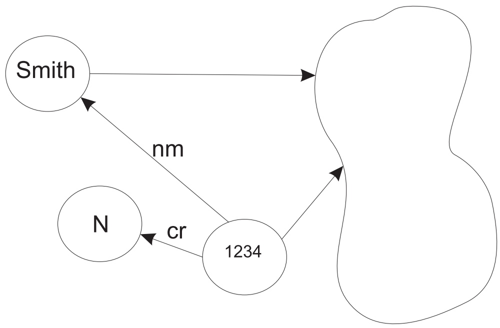

Figure 7: Small fragment of a graph database maintained by an agent A

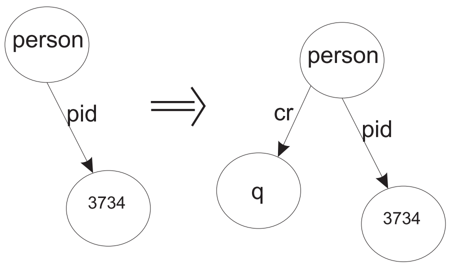

Figure 8: Production p_1

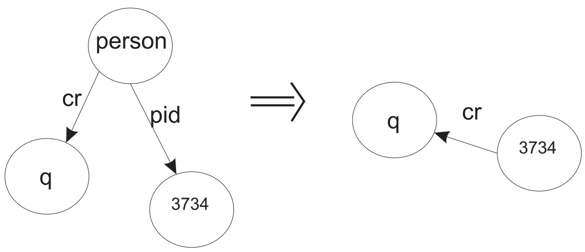

Figure 9: Production responsible for transmitting query from agent i to agent j

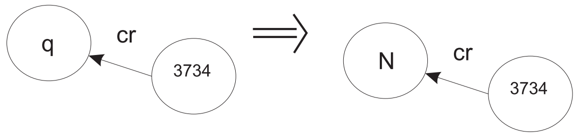

Figure 10: Production p_2

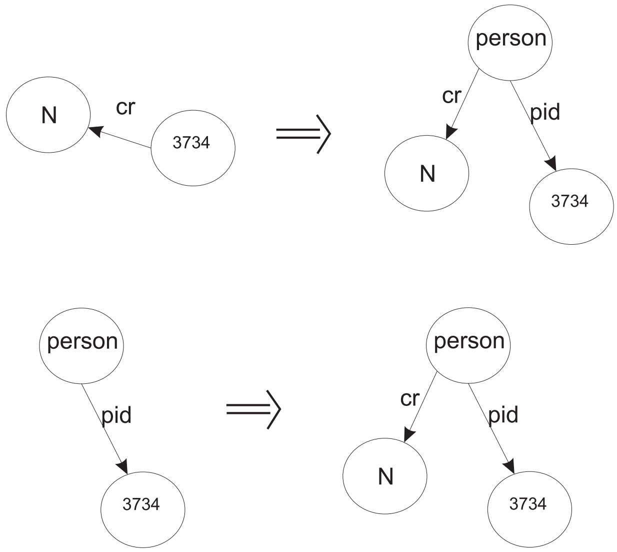

Figure 11: Productions responsible for sending requested knowledge and applying it to the graph held by agent A

appearance of many graph database management systems and the presence of more and more RDF data on the Internet. The approach presented in this paper allows for rigorous and formal description of both local maintaining and processing of knowledge and the transmission between different graph database systems. Each agent, associated with a given database maintains its own graph transformation system allowing for local manipulation and, by associating productions of the local transformation systems with global systems it also allows for the transmission of knowledge. The establishment of channels provides a formal mechanism responsible for the transmission of knowledge without imposing constraints on types of graph databases uses on either side of the channel. Thus the approach can be used on any tape of graph data.

Moreover by allowing for transmission of data only through established, predefined channels it can prevent from sending sensitive data to an unauthorized agent. The formal analysis of possible leaks of information to unauthorized entities is now required and is planned as the next step of the research. Another direction for future research is related to establishing a formal mechanism of transmitting knowledge indirectly, i.e. not between two agents at a time, but through another agent(s), which can add additional information to the knowledge passing through it.

References

[1] A. Borkowski, E. Grabska, P. Nikodem, and B. Strug. Searching for innovative structural layouts by means of graph grammars and esvolutionary optimization. In *"2nd International Structural Engineering and Construction Conference"*, pages 475–480, 2003.

[2] E. Csuhaj-Varj. Grammar systems: A short survey. In *Proceedings of Grammar Systems Week*, pages 141–157, 2004.

[3] E. Csuhaj-Varj, J Dassow, J Kelemen, and G Paun. Grammar systems: A short survey. In *Proceedings of Grammar Systems Week*, pages 141–157, 2004.

[4] J. Dassow, Gh. Paun, , and G. Rozenberg. Grammar systems. In *A. Salomaa and G. Rozenberg, editors, Handbook of Formal Languages,*, volume 2, pages 155–213, 1997.

[5] Hartmut Ehrig and Gabriele Taentzer. Graphical represenation and graph transformation. *ACM Comput. Surv.*, 31(3), 1999.

[6] E. Grabska. Theoretical concepts of graphical modelling, part one: realization of cp-graphs. *Machine Graphics and Vision*, 2(1):3–38, 1993.

[7] E. Grabska. Theoretical concepts of graphical modelling, part two: Cp graph grammars and languages. *Machine Graphics and Vision*, 2(2):149–178, 1993.

[8] E. Grabska, P. Nikodem, and B. Strug. Evolutionary methods and graph grammars in design and optimization of skeletal structuress. In *Proc. of, 11th ICE-2004*, pages 145–155, 2004.

[9] L. Kotulski. *Model wspomagania generacji oprogramowania w środowisku rozproszonym za pomoc a gramatyk grafowych.* Rozprawy Habilitacyjne. Wydawnictwo Uniwersytetu Jagiellońskiego, 2000.

[10] L Kotulski. Supporting software agents by the graph transformation systems. *LNCS*, 3993:887–890, 2006.

[11] L Kotulski, A Sedziwy, and B Strug. Heterogeneous graph grammars synchronization in cad systems supported by hypergraph representations of buildings. *Expert Systems with Applications*, 41(4):990–998, 2014.

[12] L Kotulski and B Strug. Supporting communication and cooperation in distributed representation for adaptive design. *Advanced Engineering Informatics*, 27(2):220–229, 2013.

[13] P. Nikodem and B. Strug. Graph transformations in evolutionary design. *Lecture Notes in Computer Science*, 3070:456–461, 2004.

[14] I Robinson, J Webber, and E Eifrem. *Graph databases.* O'Reilly, 2013.

[15] G. Rozenberg. *Handbook of Graph Grammars and Computing by Graph. Transformations.* World Scientific, London, 1997.

[16] B. Strug and G. Ślusarczyk. Reasoning about designs through frequent patterns mining. *Adv.Eng. Inf.*, 23:361–369, 2009.

[17] B. Strug and G. Ślusarczyk. Frequent pattern mining in a design supporting system. *Key Engineering Materials*, 450:1–4, 2011.